

CLAIMS

What is claimed is:

1 1. A method for incrementally scaling a file system, comprising:
2 adding a new file storage device to a file system having a storage
3 space comprising at least one other file storage device having a plurality
4 of directories and files stored thereon to form a new file system
5 configuration; and
6 migrating a portion of the files from said at least one other file
7 storage device to the new file storage device while hiding such migration
8 from client applications that access files from the file system so as to not
9 affect file access operations requested and performed by the client
10 applications during the migration.

1 2. The method of claim 1, wherein the portion of files that are
2 migrated from said at least one storage device to the new storage device
3 is selected such that the files are distributed across all of the storage
4 devices in the file system after the migration is completed based on a
5 relative capacity of each of the storage devices in the system.

1 3. The method of claim 1, wherein the file storage devices are
2 accessed using a file system protocol, further comprising providing a
3 storage abstraction layer between the client applications and the file

4 system protocol, said storage abstraction layer providing an interface to
5 the client applications that presents the file system as a virtual file system.

1 4. The method of claim 3, further comprising providing information
2 corresponding to the new file system configuration to the storage
3 abstraction layer.

1 5. The method of claim 3, wherein the storage abstraction layer
2 distributes new files created by the client applications across all of the
3 storage devices in the file system so as to load balance access operations
4 of the files.

1 6. The method of claim 3, further comprising:
2 filtering requests made by client applications to access a file stored
3 on the file system, said requests referencing a virtual storage location of
4 the file; and

5 remapping the file access requests that are filtered from the virtual
6 storage location to a physical location on a storage device on which the
7 file is actually stored; and

8 accessing the file through use of the file system protocol by
9 referencing the physical location of the file.

1 7. The method of claim 1, wherein migrating the files to the new
2 storage device comprises:
3 identifying a source location corresponding to a storage device and
4 directory in which each file is initially stored;
5 identifying a destination location for each file corresponding to a
6 directory on the new storage device the file is to be stored in;
7 copying each file from the source location to the destination
8 location;
9 deleting each file from its source location;
10 monitoring for any file access requests made by an client
11 application while the file is being migrated; and
12 aborting the migration of the file if a file access request is made
13 during the migration of the file.

1 8. The method of claim 7, further comprising putting any file whose
2 migration is aborted into a queue such that the migration of such file may
3 by retried at a future time.

1 9. The method of claim 7, further comprising:
2 providing a lock on each file during its migration; and
3 allowing the lock to be stolen by a client application if the client
4 application requests access to the file during its migration.

1 16. The method of claim 13, further comprising selecting (a) set(s)
2 of fragments to be migrated when a new storage device is added to the
3 system.

1 17. The method of claim 16, wherein the set(s) of fragments that
2 are selected comprise a portion of a total number of directories on all of
3 the storage devices in the file system such that after the set of fragments
4 are migrated, each storage device has a proportionate amount of
5 directories based upon its relative capacity.

1 18. The method of claim 1, further comprising:
2 providing an administrative tool that enables a user to add a new
3 storage device to the configuration of the file system; and
4 automatically selecting the portion of files to be migrated to the new
5 storage device based on the new configuration.

1 19. The method of claim 1, wherein the file system comprises a
2 virtual volume corresponding to storage space provided by at least one
3 storage device, said virtual volume including a plurality of virtual
4 directories in which virtual files may be stored and having configuration
5 data stored on the file system that maps virtual directories to physical
6 directories.

1 20. The method of claim 19, wherein the configuration
2 information comprises a master directory stored on a storage device, said
3 master directory including a plurality of subdirectories, each
4 corresponding to a respective virtual directory and having an encoded
5 pointer that points to a location on the file system where files
6 corresponding to the virtual directory are physically stored.

1 21. The method of claim 20, wherein the configuration
2 information further comprises a fragment map that identifies what storage
3 device a directory and its files are stored on based upon the fragment(s)
4 the directory is assigned to.

1 22. A method for load balancing file access on a network file
2 system having a storage space provided by a plurality of network storage
3 devices in which a plurality of files are stored, comprising:
4 partitioning the storage space into a plurality of fragments, each
5 fragment being mapped to one of said plurality of network storage
6 devices;

7 assigning files among said plurality of files to fragments such that
8 each fragment, on average, comprises a substantially equal number of
9 files;

10 migrating files among said plurality of files from network storage
11 devices on which they are initially stored to other network storage devices

1 33. The network file system of claim 32, wherein the storage
2 abstraction layer further includes an agent module running on the
3 operating system and in communication with the filter driver that maintains
4 a copy of the mapping information and forwards appropriate mapping
5 information to the filter driver.

1 34. The network file system of claim 31, wherein the operating
2 system provides the client application with access to files stored on the
3 network file system via a set of calls implemented using an NFS (Network
4 File System) protocol, and wherein the storage abstraction layer
5 comprises at least one NFS daemon that intercepts calls to the network
6 file system that reference virtual directories and virtual file names and
7 remaps the directories and file names in such calls so that they reference
8 the physical directories and file names to which the virtual directories and
9 file names correspond, based on the mapping data.

1 35. The network system of claim 34, wherein the storage
2 abstraction layer further includes an agent module running on the
3 operating system and in communication with said at least one NFS
4 daemon that maintains a copy of the mapping information and forwards
5 appropriate mapping information to said at least one NFS daemon.

1 36. The system of claim 31, wherein the virtual file system
2 comprises at least one virtual volume that is mapped to a set of physical
3 directories and files on at least one of the file storage devices, further
4 comprising:
5 a master directory stored on one of the file storage devices that
6 includes a plurality of subdirectories, each being named based on a
7 corresponding virtual directory name and including at least one file having
8 a name comprising indicia that identifies the location of the physical
9 directory on the network file system corresponding to the virtual directory
10 name.

1 37. The system of claim 36, wherein the indicia comprises a first
2 portion that identifies the fragment the files are assigned to and a second
3 portion identifying a name of the physical directory in which the files are
4 stored.

1 38. The system of claim 31, further comprising:
2 an administrative tool running on one of said at least one client
3 machines or another machine connected to the network, which enables a
4 user to define an original or new configuration of the file system and
5 provides configuration information concerning the original or new
6 configuration of the file system defined by the user to other software
7 components in the system.

